



No. 2006–04

**WHY METHODS FOR OPTIMIZATION PROBLEMS WITH
TIME-CONSUMING FUNCTION EVALUATIONS AND INTEGER
VARIABLES SHOULD USE GLOBAL APPROXIMATION
MODELS**

By Lonneke Driessen, Ruud Brekelmans, Marloes Gerichhausen,
Herbert Hamers, Dick den Hertog

January 2006

ISSN 0924-7815

Why methods for optimization problems with time-consuming function evaluations and integer variables should use global approximation models

LONNEKE DRIESSEN

Centre for Quantitative Methods BV, Eindhoven,

RUUD BREKELMANS

MARLOES GERICHHAUSEN

HERBERT HAMERS

DICK DEN HERTO¹

Tilburg University, Faculty of Economics and Business Administration,

Department of Econometrics and OR, Tilburg.

January 14, 2006

Abstract:

This paper advocates the use of methods based on *global* approximation models for optimization problems with time-consuming function evaluations and integer variables. We show that methods based on local approximations may lead to the integer rounding of the optimal solution of the continuous problem, and even to worse solutions. Then we discuss a method based on global approximations. Test results show that such a method performs well, both for theoretical and practical examples, without suffering the disadvantages of methods based on local approximations.

Keywords: approximation models, black-box optimization, integer optimization

JEL classification: C61

¹Corresponding author: Dick den Hertog (d.denhertog@uvt.nl), Tilburg University, Faculty of Economics and Business Administration, Department of Econometrics and OR, P.O.Box 90153, 5000 LE Tilburg, The Netherlands
The authors gratefully acknowledge financial support from EIT-io.

1 Introduction

In this paper we consider the following integer optimization problem:

$$\begin{aligned} \min_x \quad & F(x, s(x)) \\ \text{s.t.} \quad & x \in D, \end{aligned} \tag{1}$$

where $x \in \mathbb{Z}^p$ is a vector of integer design variables, $s(x) \in \mathbb{R}^k$ is a vector of simulation outcomes (also named responses), that depend on the value of x , F is an explicitly known, real-valued objective function, and $D \subset \mathbb{Z}^p$ is an explicitly known bounded region of feasible design points. The relationship between $s(x)$ and x is implicitly known and evaluating the value of $s(x)$ for a certain x is time consuming. This relationship could be defined in a black-box simulation model, for example. We assume that Problem (1) is deterministic and that the vector x consists of a limited number of integer or discrete variables.

In engineering design problems integer or discrete variables often arise, for example the number of fins on a heatsink. Applications of the above optimization problem can also be found in logistic process design and analysis. In the last decade computer simulations are widely used in both fields. Simulation times are often high, sometimes even hours or a full day. Each scenario corresponds with a specific setting of the design variables. Due to time restrictions, only a limited number of scenarios can be evaluated and designers are confronted with the problem of deciding which scenarios to run. The crucial question becomes how to find the best possible design with a minimum number of simulations. There is no explicit information about the functional relationship between the design variables and the resulting process characteristics arising from the black-box computer model.

A broad range of distinct derivative free optimization approaches, mainly for continuous problems, can be distinguished. Schoofs *et al.* (1994), Toropov (1999), and Brekelmans *et al.* (2005) are examples of iterative optimization methods in which local polynomial approximations guide the search for an optimal design. Dennis and Torczon (1994), Torczon (1992), and Booker *et al.* (1999) describe (extensions of) a direct search method named pattern search. The pattern search framework consists of a search step and a polling step. The latter ensures convergence to a local optimum. The search step can involve any search strategy, from doing nothing at all to building global approximation models to speed up the search for an optimum. For an overview of derivative free optimization methods see Lewis *et al.* (2001).

Besides iterative algorithms based on local approximations, also non-iterative algorithms based on global approximations are proposed in the literature. See e.g. , Sachs *et al.* (1989), Balabanov *et al.* (1999), and Den Hertog and Stehouwer (2002). In these methods explicit global approximations for the functional relationships between the design variables and the resulting responses are constructed using Design of (Computer) Experiments techniques to create a simulation scheme. The advantage of global approximation methods is that global insight is obtained in the behavior of these responses. Moreover, when the objective function F changes or the design area D decreases, no new simulations have to be carried out to find a new optimum, since the global approximations still can be used. The disadvantage of this approach, however, is that for

high-dimensional design problems, the number of simulations required to obtain good global approximations becomes too high.

A good way to circumvent this high number of initial simulations is to construct the global approximations iteratively. Starting from a small set of initial simulations, global approximating models are created and refined in subsequent steps of the algorithm. Jones *et al.* (1998), Björkman and Holmström (2000), and Regis and Shoemaker (2005) describe such methods. See also Jones (2001) for a good review. The above described design optimization methods mainly focus on continuous design variables. Several papers describe optimization methods based on local approximations for solving Problem (1), in which the design variables are integers. See e.g. Bremicker *et al.* (1990), Loh and Papalambros (1991b), Loh and Papalambros (1991a), Abspoel *et al.* (2001), Gijsbers (2003), and Jacobs (2004). These local approximations, based on evaluations around the current iterate, are trusted only in a certain neighborhood of the current iterate. This neighborhood is often called the trust region. The next iterate is determined by substituting the local approximations and by optimizing over the trust region. For a formal description see Jacobs (2004). Disadvantage of these methods is that the use of local approximation models can easily result in ending up in the neighborhood of the continuous optimum (which is in general not the same as the integer optimum), or even worse.

Other approaches which are used especially in the field of discrete event simulation, are for example local search, simulated annealing, and tabu search (e.g. Glover *et al.* (1996)). These methods, however, assume that one simulation run can be performed quickly and require many runs.

We insist on using global instead of local approximation models for Problem (1), in which the design variables are integers, as the use of local approximation models can easily result in ending up in the neighborhood of the continuous optimum (which is in general not the same as the integer optimum), or even worse. Another disadvantage of methods based on local approximations is that, even for problems with continuous variables, they converge to a local optimum. Methods based on global approximations have a better global overview of the responses over the whole design space and hence, a larger chance to find the global optimum.

In this paper we propose a solution method based on global approximation models to solve Problem (1). The method iteratively creates Kriging approximation models for the different responses. These models are based on the simulation data obtained so far. Using these models an explicit optimization problem is solved to obtain a promising design for the next simulation. The initial approximation models are based on a maximin Latin Hypercube Design (LHD) of computer experiments.

The remainder of this paper is organized as follows. In Section 2 we discuss the disadvantages of using local approximation models. Section 3 elaborates on our proposed solution approach. Numerical results are presented in Section 4. In this section we also compare to OptQuest (Glover *et al.* (1999)), a commercial optimization package often used in discrete event simulation. Section 5 concludes.

2 Drawbacks of methods using local approximations

Using optimization methods based on local approximations for solving Problem 1 has several disadvantages. This section will discuss them with the help of two examples.

For integer linear programming (ILP) problems it is well-known that solving the continuous version of an ILP and then rounding the solution to an integer solution, often yields a bad solution. Therefore, special algorithms are developed for ILP problems. We now want to argue that for the integer Problem (1) methods based on local approximations may lead to the integer rounding of the optimal solution of the continuous problem, and even to worse solutions.

Suppose that our original Problem (1) is the following ILP:

$$\begin{aligned}
 & \max_{x_1, x_2} && (1 + 2\varepsilon)x_1 + x_2 \\
 & \text{s.t. } x_1 &\geq & 0, \\
 & & x_2 &\geq 0, \\
 & (1 + \varepsilon)x_1 + x_2 &\leq & 10, \\
 & x_1 &\in & \mathbb{Z}, \\
 & x_2 &\in & \mathbb{Z},
 \end{aligned} \tag{2}$$

where $s(x) = (1 + 2\varepsilon)x_1 + x_2$, $F(x, s(x)) = s(x)$, and $\varepsilon > 0$ is a small constant. Moreover, suppose that we use local linear approximations, which have a perfect fit in this case, and that our trust region is the unit box $\|x - x^k\|_\infty \leq 1$, in which x^k is the current iterate. It is easy to see that e.g. starting in $(0, 0)$, we will get the following iterate sequence: $(1, 1)$, $(2, 2)$, $(3, 3)$, $(4, 4)$, $(5, 4)$, $(6, 3)$, $(7, 2)$, $(8, 1)$, $(9, 0)$. Hence, the final solution is indeed an integer rounding of the optimal solution $(\frac{10}{1+\varepsilon}, 0)$ of the continuous problem, whereas the integer optimal solution is $(0, 10)$. In Figure 2.1 all solution paths are shown. It appears that all feasible starting solutions end up in the wrong solution $(\frac{10}{1+\varepsilon}, 0)$, except when we start in $(0, 9)$, which is very close to the integer optimal solution.

We now reduce the trust region to the unit sphere $\|x - x^k\|_2 \leq 1$. In Figure 2.2 the solution paths are shown for this case. The algorithm can converge to many different solutions, which are even worse than the integer rounding of the optimal solution of the continuous problem, $(9, 0)$.

One would expect that enlarging the trust region will decrease the number of iterations. That is not necessarily always the case. Suppose that the trust region is the following ellipsoid: $\frac{1}{4}(x_1 - x_1^k)^2 + (x_2 - x_2^k)^2 \leq 1$. Then starting in $(0, 0)$ leads to the following solution path: $(0, 0)$, $(2, 0)$, $(4, 0)$, $(6, 0)$, $(8, 0)$, $(9, 0)$, containing five steps. When we use the larger trust region $|x_1 - x_1^k| \leq 2, |x_2 - x_2^k| \leq 1$, the solution path becomes $(0, 0)$, $(2, 1)$, $(4, 2)$, $(6, 3)$, $(7, 2)$, $(8, 1)$, $(9, 0)$, which contains six steps.

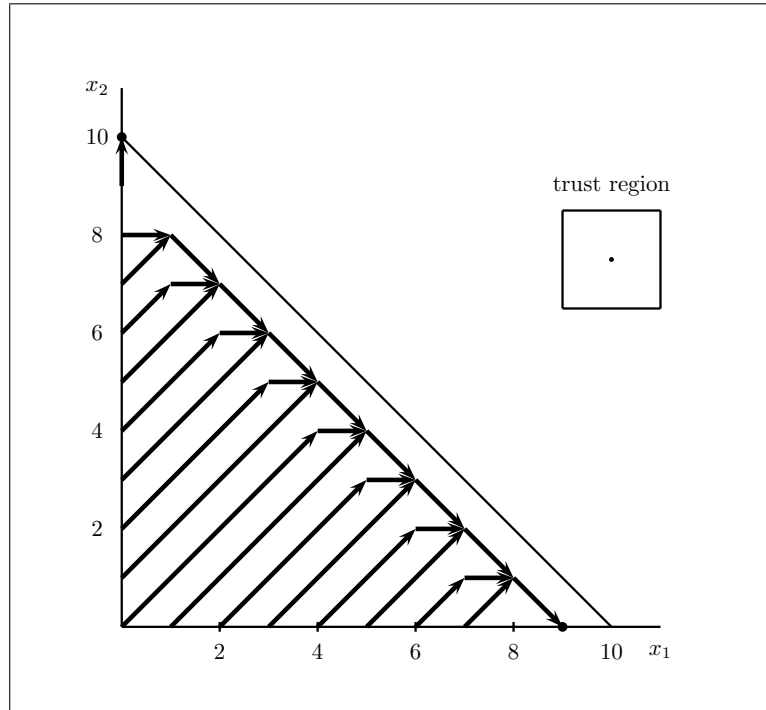


Figure 2.1: Solution paths for Problem (2) in case of a unit box as trust region.

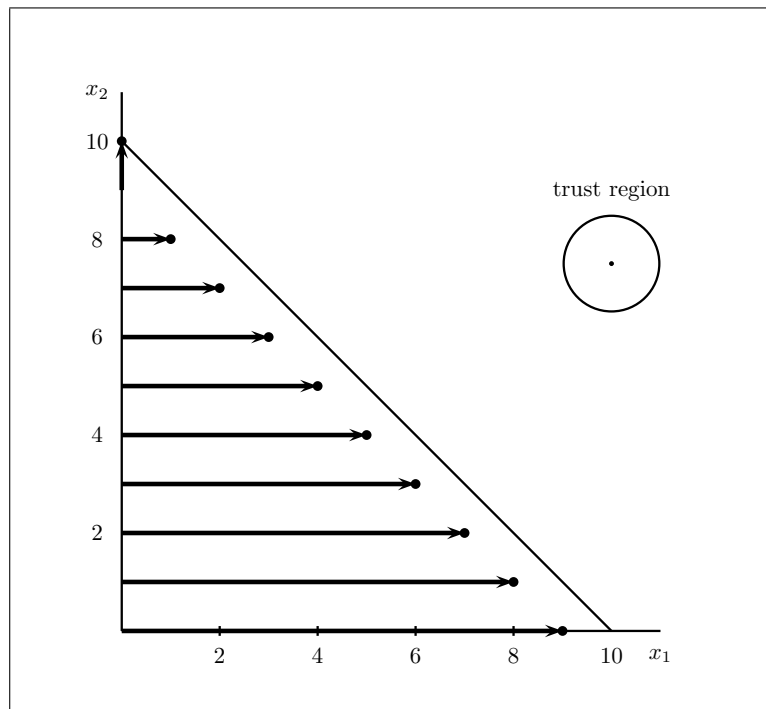


Figure 2.2: Solution paths for Problem (2) in case of a unit sphere as trust region.

Let us now look at the following slightly different problem:

$$\begin{aligned}
& \max_{x_1, x_2} && (1 + \varepsilon)x_1 + x_2 \\
& \text{s.t. } x_1 && \geq 0, \\
& && x_2 \geq 0, \\
& && (2 + \varepsilon)x_1 + x_2 \leq 10, \\
& && x_1 \in \mathbf{Z}, \\
& && x_2 \in \mathbf{Z},
\end{aligned} \tag{3}$$

where $s(x) = (1 + \varepsilon)x_1 + x_2$, $F(x, s(x)) = s(x)$, and $\varepsilon > 0$ is a small constant. Suppose the trust region is the unit box. Figure 2.3 shows the solution paths for this case. Again, the algorithm may end in different non-optimal solutions. Note that in this case the integer rounding of the continuous optimal solution is optimal, but only a few starting solutions close to the optimal solution, lead to this optimal solution.

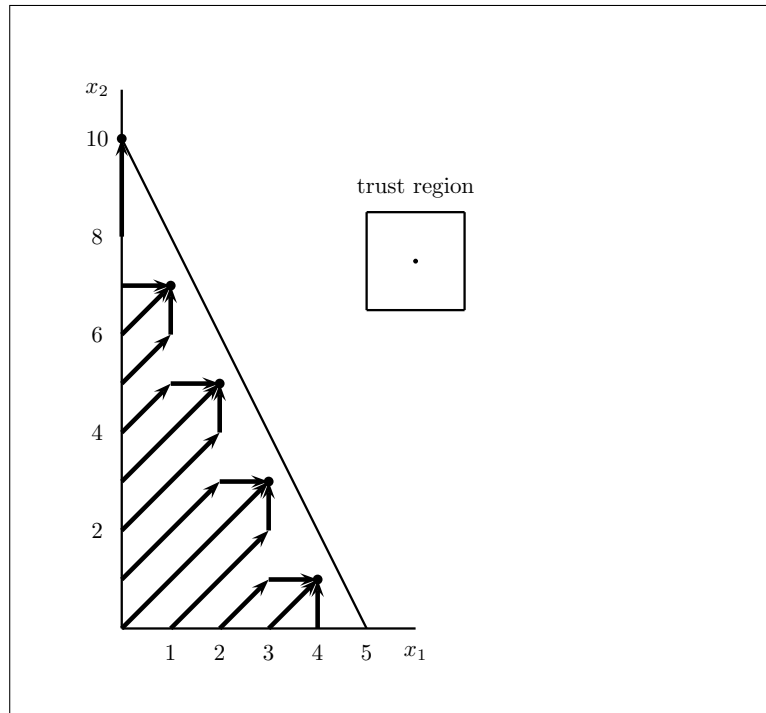


Figure 2.3: Solution paths for Problem (3) in case of a unit box as trust region.

Let us now enlarge the trust region as shown in Figure 2.4. This figure also shows the solutions paths. For several starting points the final solution is even worse than in Figure 2.3, which is based on a smaller trust region. For example, starting in $(0, 0)$ leads to $(3, 3)$ in Figure 2.3, and to the worse solution $(4, 1)$ in Figure 2.4.

Our final conclusion is that methods based on local approximations may end up in the wrong integer rounding of the continuous optimal solution. They may even end up in many different non-optimal solutions, depending

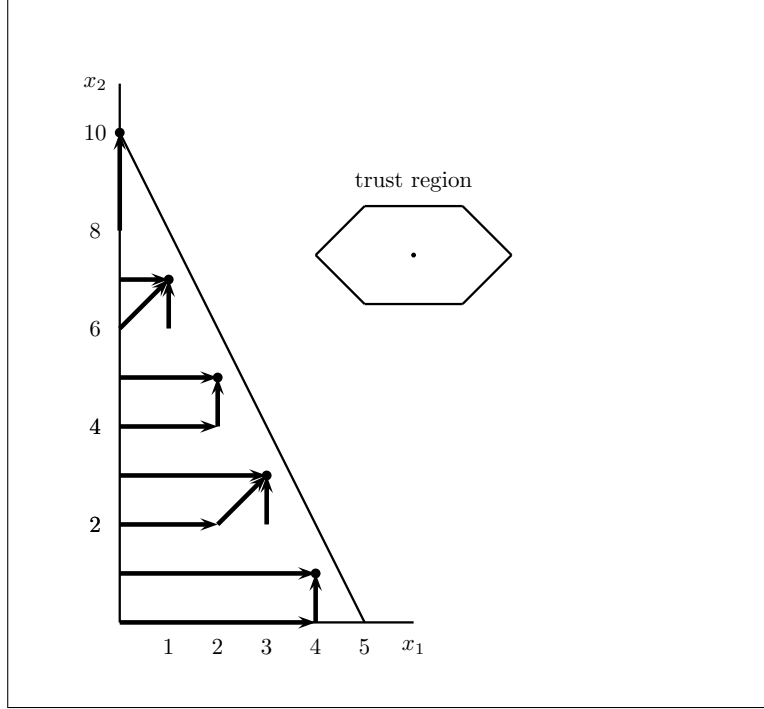


Figure 2.4: Solution paths for Problem (3) in case of an enlarged trust region.

on the starting solution. Moreover, enlarging the trust region may increase the number of iterations needed, and may lead to worse solutions. These conclusions are based on two-dimensional examples; for higher dimensional problems the effects can be even more disastrous.

We therefore plead for the use of global approximation methods in case of optimization problems when the number of variables is not too high. Building reliable global approximation models for a high-dimensional space requires a very large initial set of simulations. Hence, in case of many variables, global approximation models are out of reach and building local approximation models is the only alternative. In that case it is advisable to use a multi-start method to avoid getting stuck in local optima. In the next sections we propose and analyze an example of a method based on global approximations.

3 A method using global approximations

In the former section we discussed the disadvantages of local approximation methods for solving Problem (1). In this section we propose a method based on *global* approximation models. We will refer to this method as GAM (Global Approximation Method).

In this method the search for an optimal x is guided by the global approximation models \hat{s} . These models

replace the unknown relationships s in Problem (1). Hence, in each iteration we solve

$$\begin{aligned} \min_x \quad & F(x, \hat{s}(x)) \\ \text{s.t.} \quad & x \in D \setminus Q, \end{aligned} \tag{4}$$

where Q indicates the set of already simulated designs.

Figure 3.5 presents a flowchart of GAM. The first step is to generate a Design of Computer Experiments of d points for the design variables x . For this initial set of design points, $\{x_1, \dots, x_d\}$, simulations are carried out. This results in a set $P = \{(x_1, s(x_1)), \dots, (x_d, s(x_d))\}$ of design-response combinations. The vector s consists of k different simulation outcomes. For each element $s_i(x)$, $i = 1, \dots, k$, an approximation model $\hat{s}_i(x)$ can be constructed based on the data set P . Each $\hat{s}_i(x)$ approximates for any value $x \in D$ the corresponding value of $s_i(x)$. The approximation models are then used to locate the most promising value for x . Hence, as a next step we solve Problem (4), resulting in an optimum x^* . Global optimization solvers can be used to solve Problem (4), by defining the set of constraints needed to exclude Q from D . We simulate x^* to obtain $s(x^*)$. Note that this simulation is the most time consuming operation of the iteration. If the approximation models are adequate, the difference between the predicted values $\hat{s}(x^*)$ and the real values $s(x^*)$ is small. The quality of the approximation models in general improves as more iterations have been carried out. The next step is to check the stopping conditions. These stopping conditions could include a maximum number of simulations, a maximum amount of simulation time, and a satisfaction level with respect to objective function value, for example. If the stopping conditions are satisfied, we are done. Otherwise we add the new combination $(x^*, s(x^*))$ to the set P of points on which the approximation models are based and start the next iteration of constructing approximation models, etc. Having described the main flow of GAM, there are two elements that deserve some more attention: the initial set of simulations and the approximation models. They are discussed below.

Initial set of simulations

For building global approximation models at least a small initial set of simulations is needed. The optimal size of this set depends on the type of the approximation model, the number of input variables, and possibly on prior knowledge about the nonlinearity of the response to be approximated. The problem of choosing the initial set is called Design of Experiments (DOE) (Montgomery (1984)). Many different DOE schemes exist. Well known classical designs are the full / fractional factorial designs, Box-Behnken designs, central composite designs, and D-optimal designs. Another class of designs, Design of Computer Experiments (DoCE), has been specially developed for use with computer simulations (see Koehler and Owen (1996)). Examples are space-filling designs, random designs, orthogonal arrays, and LHDs.

In DoCE a desirable property of initial schemes is space-fillingness. One has to choose the points such that as much information as possible is captured. Intuitively this is the case when the points are spread throughout the feasible region as evenly as possible, i.e., the scheme is space-filling. Hereby we assume

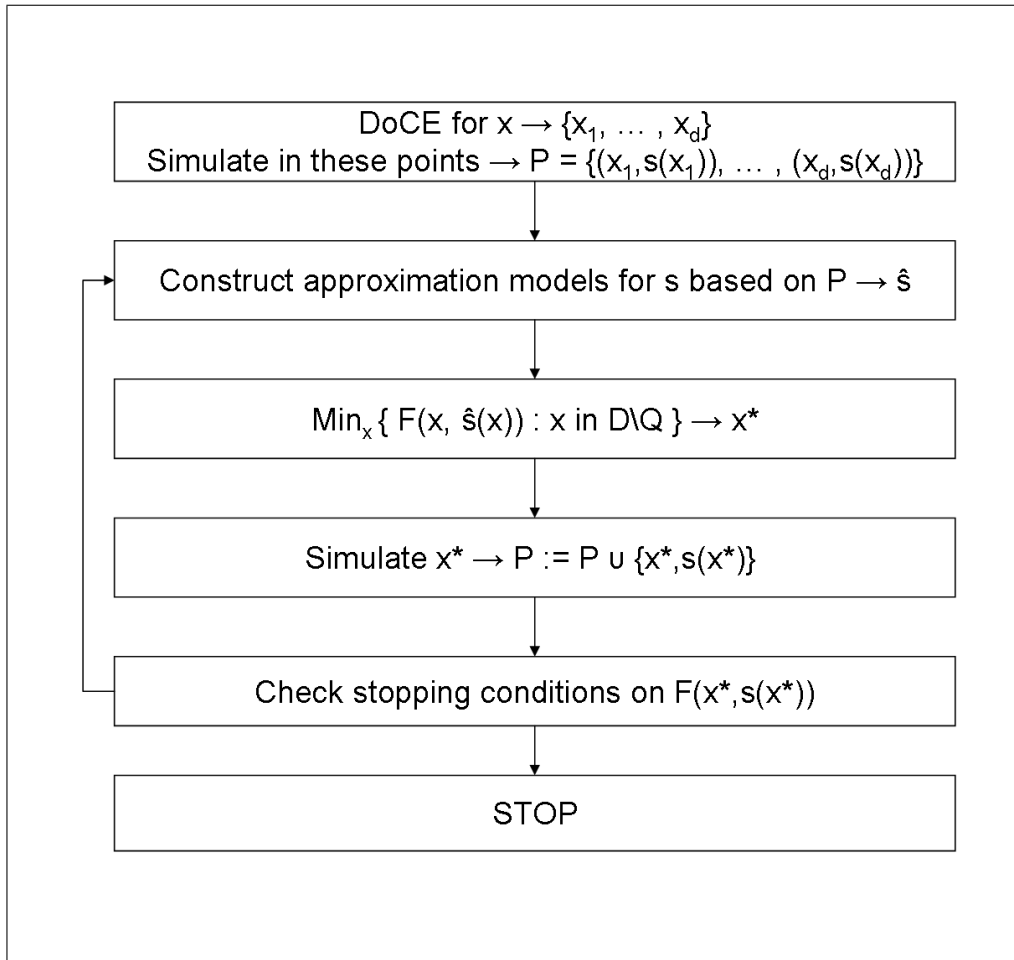


Figure 3.5: Flowchart of GAM.

that no information is available about the function to be approximated. Another desirable property for a scheme is to be non-collapsing, which implies that for each input dimension separately the variable values are all distinct and well spread. Stehouwer and Den Hertog (1999) developed an approach for generating space-filling non-collapsing schemes for continuous variables. This method searches for the most space-filling scheme within the class of LHDs. It extends the approach presented by Morris and Mitchell (1995). The approach iterates over two steps. The first step constructs a random LHD. The second step searches for the best possible LHD with respect to space-fillingness. This search is performed using Simulated Annealing and the objective function guiding it is (Morris and Mitchell (1995))

$$\varphi_g(D) = \left(\sum_{j=1}^m J_j r_j^{-g} \right)^{\frac{1}{g}}, \quad (5)$$

where g is a positive integer, r_j is the j^{th} element of a distance list (r_1, \dots, r_m) in which the elements are the distinct values of the inter-point distances, sorted from the smallest to the largest, and J_j is the number of pairs of points in D separated by distance r_j .

The algorithm of Stehouwer and Den Hertog (1999) has been extended to be able to deal with integer variables. In this case the LHD property is only valid when there are less initial points than integer levels in a certain dimension. The LHD property becomes a soft constraint for the integer variant: collapsing schemes get a high penalty in the objective function. The space-filling property remains valid in all cases. Figure 3.6 illustrates this.

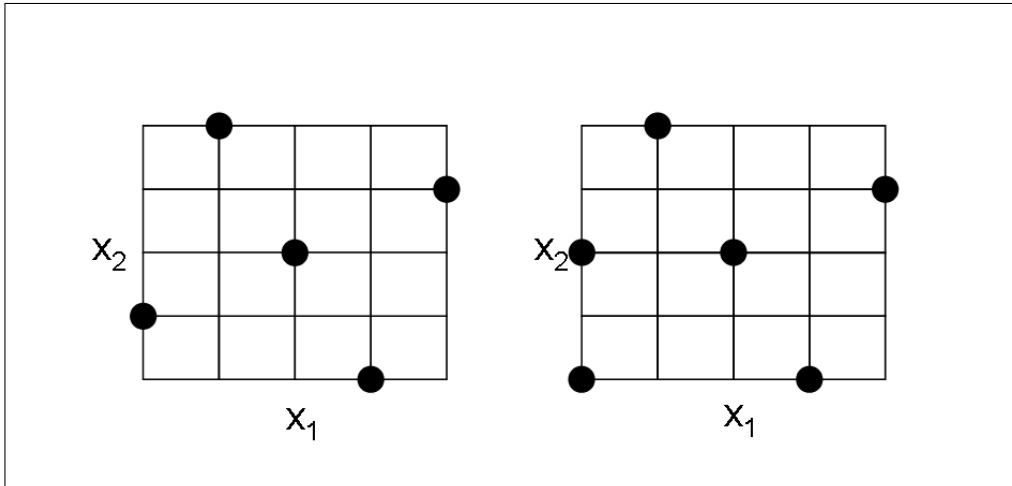


Figure 3.6: DoCE for a 2D design space. Both x_1 and x_2 have 5 possible integer levels. The picture to the left shows a DoCE with 5 design points, the picture to the right shows a DoCE with 6 design points. In the latter picture it is impossible to fully maintain the LHD property.

Approximation models

The choice of the approximation model type is not an easy one. Depending on the nonlinearity of the response a more sophisticated approximation model is required. To be able to deal with nonlinear behavior

as well we chose to use the Kriging model as approximation model, which was introduced as approximating functions for deterministic computer simulations by Sachs *et al.* (1989). In a Kriging model the response function $s_i(x)$ is treated as a realization of a stochastic process, just like in linear regression. A difference between a Kriging model and a regression model is that the errors in the latter model are assumed to be independent, whereas in the Kriging model they are assumed to be correlated. The correlation between two data points is larger as they lie closer together. Another difference is that Kriging functions are interpolating. In fact the Kriging function is the Best Linear Unbiased Predictor. The resulting Kriging approximation function is of the following form

$$\hat{s}_i(x) = c_{i0} + \sum_{k=1}^d c_{ik} e^{-\sum_{j=1}^p \theta_{ij} \|x_j - x_j^k\|^{p_{ij}}}, \quad (6)$$

in which the model coefficients c_{i0} , c_{ik} , θ_{ij} , and p_{ij} depend on the set of data points on which the Kriging model is fitted. The coefficients θ_{ij} and p_{ij} are obtained by solving a Maximum Likelihood Estimator problem. Often the choice is made to keep $p_{ij} = 2$ fixed to obtain smooth Kriging models. Finding the coefficients of a Kriging model is computationally expensive, but much less expensive than a black-box simulation. We use a pattern search technique to find the coefficients θ_{ij} and keep $p_{ij} = 2$ fixed. More details about Kriging models, their underlying assumptions, and how to calculate the model coefficients can be found in Sachs *et al.* (1989).

For problems with a large variable range the described method is insufficient and requires extension. A well known issue for methods based on global approximations for continuous optimization problems, is the need for so called geometry improving iterations in the optimization process. Without such mechanism the optimization process can easily get stuck proposing only candidates that are in the neighborhood of the best solution so far, as the global approximation model probably predicts a new optimum close to that best solution so far. To ensure global exploration, also proposals based on the location of a point in the design space instead of on its objective value are generated. Points located far away from already simulated points are most interesting in this respect as they apparently are located in relatively unexplored areas of the design space. Regis and Shoemaker (2005) describe an elegant way to ensure sufficient global exploration. They add an extra set of constraints that ensures that the next proposal is far enough away from already simulated points. The definition of what is far enough away is altered in each iteration and ranges from no restriction at all (local search) to requiring a large distance to already simulated points (global search).

For integer optimization problems with a large variable range, the above described need for geometry improvement arises, as these variables behave almost like continuous variables.

4 Test results

In this section we discuss the results obtained on a set of academic test cases and compare these results with OptQuest, a commercial optimization package often used in discrete event simulation. Thereafter we

present our results on two practical applications.

As a first test of the performance of GAM, we took the two linear problems described in Section 2. GAM found the integer optimum in 6 iterations for Problem (2) and in 7 iterations for Problem (3).

As a next test, we used the set of academic problems listed in Table 4.1. In each problem the vector of responses consists of one single response, and $F(x, s(x)) = s(x)$. To create a discrete version of the problems in Table 4.1, we introduced a stepsize and allowed for solutions ranging from lower to upper bound with steps of a size equal to the given stepsize.

Problem B is the Branin test function, Problem S is the Six-hump-camel-back test function, Problem R and Problem RL are instances of the generalized Rosenbrock function, and Problem L is the Levy function. The results on these test problems are listed in Table 4.2. We use the OptQuest Solver Free Demo Version 1.5 (OptQuest Version 4.0.5.2), with settings auto stop = off, database size = 10000, objective precision = 0.000001 and variables precision = 0.000001. We run the solver for different values of the number of iterations. Note that this demo version can only locate one optimum, hence for Problem S OptQuest finds only one of the two optima. In our implementation of GAM we use $g = 3$ in (5). Instead of using an NLP solver for Problem (4), we iterate through all feasible solutions and choose the best one.

We can conclude from the results in Table 4.2 that GAM finds the global optimum quickly and performs well in comparison to OptQuest. Moreover, Figure 4.7 shows that GAM often finds large improvements almost immediately after the DoCE phase has been completed. In a situation in which each simulation takes hours and there is a restriction on the number of simulations that can be carried out, it is very important to find large improvements as quickly as possible. Taking problem L as example and assuming a simulation time of 1 hour, the total time involved with GAM is 191 seconds GAM calculation time plus 20 hours simulation time, whereas OptQuest would need 351 hours of simulation time. The larger the simulation time, the larger the difference in solution time becomes. Problem X is the only problem for which Optquest performs better.

Next we consider two practical applications, a location problem and a closed queuing network problem. The location problem is the problem of deciding which subset of the set of possible distribution centers should be opened and how the client locations should be assigned to the open distribution centers, such that the total of setup cost and transport cost is minimized. In the context of black-box optimization we regard the choice of which distribution centers should be opened as the black-box optimization problem. Hence, with C distribution centers there are 2^C possible solutions. The calculation of total costs, including the optimal assignment of client locations to distribution centers that are open, is carried out in a black-box simulation model. In Problem LocS, the small location problem, there are 5 available distribution centers. In Problem Loc, the larger location problem, there are 10 available distribution centers.

Closed queuing networks arise for example in manufacturing units attempting to maintain a constant level of work in process, or busy computer networks for which it is assumed that a new job enters the network as soon as another job leaves. Such systems, in which a constant number of jobs is present, may be modelled

	$s(x)$	p	Bounds	S.s.	# Scen.	Opt. F
B	$[x_2 - 5.1(\frac{x_1}{2\pi})^2 + \frac{5x_1}{\pi} - 6]^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	$x_1:[2\ 10]\ x_2:[-2\ 4]$	0.1	4941	0.403
S	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$x_1:[-1.5\ 1.5]\ x_2:[-1\ 1]$	0.1	651	-1.030
Q	$x_1^2 + (x_2 - 2)^2 + 2x_3$	3	$x_{1..3}:[-10\ 10]$	1	9261	-20.000
R	$\sum_{i=1}^4 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	5	$x_{1..5}:[-5\ 5]$	1	161051	0.000
L	$\sin^2 3\pi x_1 + (x_6 - 1)(1 + \sin^2 2\pi x_6) + \sum_{i=1}^5 ((x_i - 1)^2(1 + \sin^2 3\pi x_{i+1}))$	6	$x_{1..5}:[-4\ 4]\ x_6:[-5\ 3]$	1	532441	-6.000
X	$\sum_{i=1}^7 (x_i^2 - 2\sin(2x_i))$	7	$x_{1..7}:[-3\ 3]$	1	823543	-5.730
RL	$\sum_{i=1}^5 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	6	$x_{1..6}:[-5\ 5]$	1	1771561	0.000

Table 4.1: Problem characteristics of academic test problems. Column S.s. denotes the stepsize, and column # Scen. denotes the number of possible scenario's.

	Total # simulations GAM	Total # simulations OptQuest
B	8	674
S	11, 34 (*)	55
Q	14	311
R	31	290
L	20	351
X	65	38
RL	67	144

Table 4.2: Results on academic test problems.

(*) This function has two global optima.

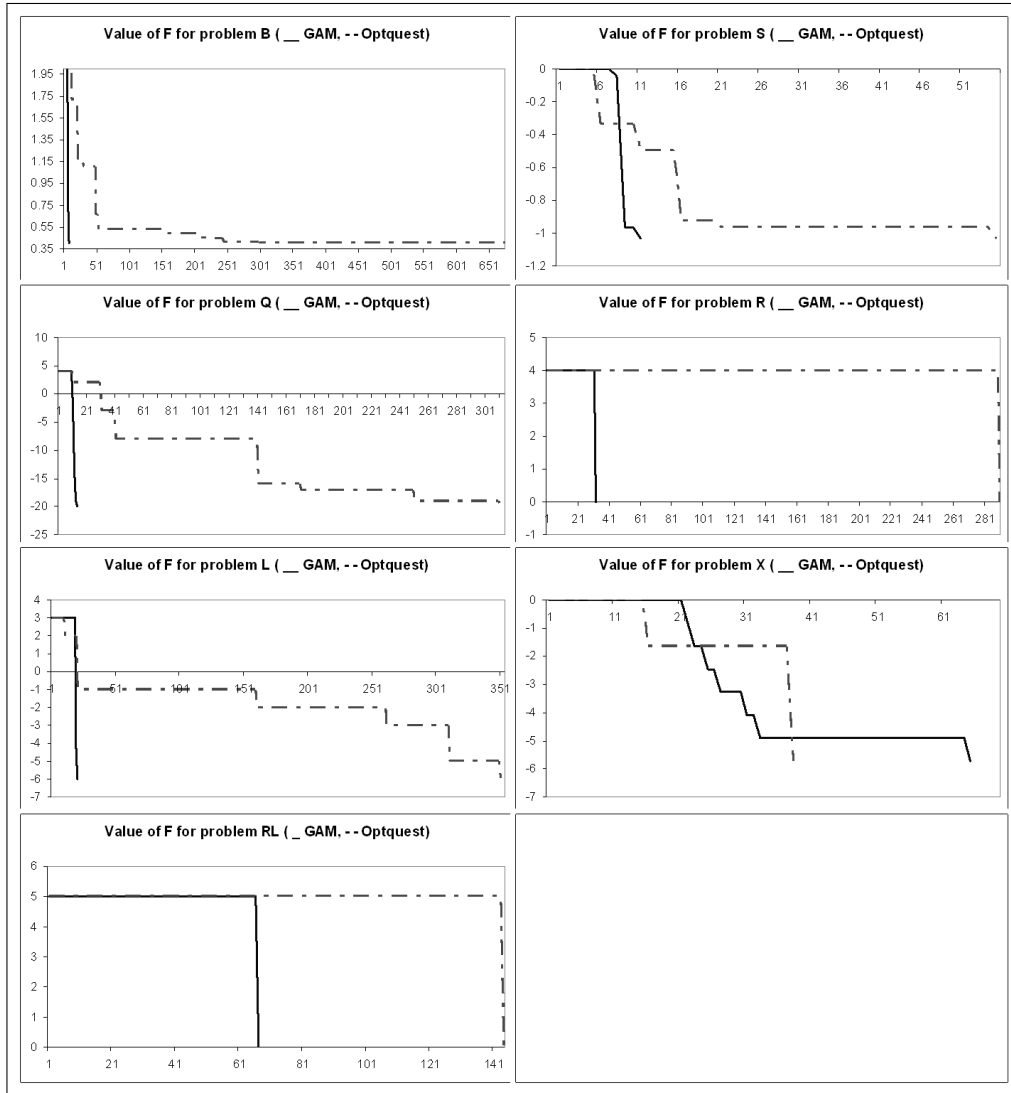


Figure 4.7: Subsequent iterations towards optimum for GAM and OptQuest.

	$s(x)$	p	Bounds	S.s.	# Scenario's	Optimal F
LocS	small location	5	$x_{1..5}:[0\ 1]$	1	32	449215.3
Loc	location	10	$x_{1..10}:[0\ 1]$	1	1024	471500
QueuingS	small queuing	3	$x_{1..3}:[-2\ 2]$	1	125	-20001
Queuing	queuing	5	$x_{1..5}:[-2\ 2]$	1	3125	-0.804162

Table 4.3: Problem characteristics of practical applications.

	Total # simulations GAM
LocS	6
Loc	27
QueuingS	6
Queuing	19

Table 4.4: Results on practical applications.

as closed queuing networks. In the small queuing problem we consider a flexible manufacturing system in which at all times 10 parts are present. There are 3 machines. Each part begins by having operation 1 done at machine 1. Then, with probability 0.75 the part has operation 2 processed on machine 2, and with probability 0.25 the part has operation 2 processed on machine 3. Once a part completes operation 2, it leaves the system and is immediately replaced by another part. Each machine can be purchased at 5 different machine rates. The fastest rate costs most. The optimization problem then becomes to choose the machine rates resulting in the highest number of service completions (in parts per minute) at lowest cost. Given the machine rates, a black-box simulation model is used to calculate the total cost (in terms of machine purchase prices and service completions). There are 125 different scenario's. In the larger queuing network there are 15 parts always present in the system and 5 machines that can be purchased at 5 different speeds, resulting in 3125 scenario's.

Problem characteristics and results for the location and queuing network problems are listed in Table 4.3 and Table 4.4. For all four problems GAM finds the optimum in very few iterations.

The above results show that GAM, a rather basic method based on global approximation methods, is very effective for integer problems with a discrete nature. Applying GAM to integer problems with a large variable range for the design parameters we find the behavior we described at the end of Section 3: GAM finds a reasonably good point rather quickly and then lingers around in the neighborhood of this point. For these kind of problems GAM should be extended with a geometry safeguard mechanism, as described in Section 3.

5 Conclusion

We have shown that in case of integer design variables, optimization methods based on local approximation models risk to end up in the neighborhood of the continuous optimum or even worse, instead of in the integer optimum. Therefore, we believe that it is better to use global approximation models instead for integer design problems. We have shown that an optimization method based on global Kriging approximation models performs very well on a set of test problems with integer design variables. In case of expensive or time consuming simulations this method outperforms the optimization method OptQuest.

References

- S.J. ABSPOEL, L.F.P. ETMAN, J. VERVOORT, R.A. VAN ROOIJ, A.J.G. SCHOOF, AND J.E. ROODA (2001). Simulation based optimization of stochastic systems with integer design variables by sequential multipoint linear approximation. *Structural and Multidisciplinary Optimization*, 22(2):125–138.
- V. BALABANOV, A.A. GIUNTA, O. GOLOVIDOV, B. GROSSMAN, W.H. MASON, L.T. WATSON, AND R.T. HAFTKA (1999). Reasonable design space approach to response surface approximation. *Journal of Aircraft*, 36(1):308–315.
- M. BJÖRKMAN AND K. HOLMSTRÖM (2000). Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering*, 1:373–397.
- A.J. BOOKER, J.E. DENNIS, P.D. FRANK, D.B. SERAFINI, V. TORCZON, AND M.W. TROSSET (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, 17(1):1–13.
- R.C.M. BREKELMANS, L.T. DRIESSEN, H.J.M. HAMERS, AND D. DEN HERTOOG (2005). Constrained optimization involving expensive function evaluations: a sequential approach. *European Journal of Operational Research*, 160(1):121–138.
- M. BREMICKER, P.Y. PAPALAMBROS, AND H.T. LOH (1990). Solution of mixed-discrete structural optimization problems with a new sequential linearization algorithm. *Computers & Structures*, 37(4):451–461.
- D. DEN HERTOOG AND H.P. STEHOUWER (2002). Optimizing color picture tubes by high-cost non-linear programming. *European Journal of Operational Research*, 140(2):197–211.
- J.E. DENNIS AND V. TORCZON (1994). Derivative-free pattern search methods for multidisciplinary design problems. In *Proceedings of the Fifth AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 922–932. MCB University Press, Panama City Beach.
- J.A.A. GIJSBERS (2003). Integer simulation-based optimization: a new multipoint approximation approach with probabilistic filter acceptance and intermediate design variables. *Systems engineering report se-420398*, Eindhoven University of Technology.

- F.W. GLOVER, J.P. KELLY, AND M. LAGUNA (1996). New advances and applications of combining simulation and optimization. In *Proceedings of the 1996 Winter Simulation Conference*, edited by J. Charnes et al., 144–152. Coronado.
- F.W. GLOVER, J.P. KELLY, AND M. LAGUNA (1999). New advances for wedding optimization and simulation. In *Proceedings of the 1999 Winter Simulation Conference*, edited by P.H. Farrington et al., 255–260. Phoenix.
- J.H. JACOBS (2004). *Performance quantification and simulation optimization of manufacturing flow lines*. PhD thesis, Eindhoven University of Technology.
- D.R. JONES (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383.
- D.R. JONES, M. SCHONLAU, AND W.J. WELCH (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.
- J.R. KOEHLER AND A.B. OWEN (1996). Computer experiments. In *Handbook of Statistics*, edited by S. Ghosh and C.R. Rao, volume 13, 261–308. Elsevier Science B.V., Boston.
- R.M. LEWIS, V. TORCZON, AND M.W. TROSSET (2001). Direct search methods: then and now. In *Numerical Analysis 2000*, volume 4, 191–207. Elsevier, North Holland.
- H.T. LOH AND P.Y. PAPALAMBROS (1991a). Computational implementation and tests of a sequential linearization algorithm for mixed-discrete nonlinear design optimization. *Journal of Mechanical Design*, 113(3):335–345.
- H.T. LOH AND P.Y. PAPALAMBROS (1991b). A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems. *Journal of Mechanical Design*, 113(3):325–334.
- D.C. MONTGOMERY (1984). *Design and Analysis of Experiments*. John Wiley & Sons, New York, 2nd edition.
- M.D. MORRIS AND T.J. MITCHELL (1995). Exploratory designs for computer experiments. *Journal of Statistical Planning and Inference*, 43:381–402.
- R.G. REGIS AND C.A. SHOEMAKER (2005). Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31(1):153–171.
- J. SACHS, W.J. WELCH, T.J. MITCHELL, AND H.P. WYNN (1989). Design and analysis of computer experiments. *Statistical Science*, 4:409–435.
- A.J.G. SCHOOF, P.J.M. ROOZEN-KROON, AND D.H. VAN CAMPEN (1994). Optimization of structural and acoustical parameters of bells. In *Proceedings of the Fifth AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. MCB University Press, Panama City Beach.

- H.P. STEHOUWER AND D. DEN HERTOOG (1999). Simulation-based design optimisation: methodology and applications. In *Proceedings of the First ASMO UK / ISSMO Conference on Engineering Design Optimization*, 349–355. MCB University Press, Ilkley.
- V. TORCZON (1992). PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines. *Technical report 92-09*, Rice University, Department of Computational and Applied Mathematics.
- V.V. TOROPOV (1999). Multipoint approximations based on response surface fitting: a summary of recent developments. In *Proceedings of the First ASMO UK / ISSMO Conference on Engineering Design Optimization*, 371–380. MCB University Press, Ilkley.